REMARKS

Claims 1-8, 10-17, 19 and 22 stand as previously presented. Claim 21 has been cancelled. Claims 9, 18 and 20 have been amended. Claim 23 has been added.

Claims 1-22 were considered in the Office Action.

Claims 1-11, 16, 17 and 20-22 stand rejected under 35 U.S.C. 102(e) as being anticipated by Bugnion, U.S. Patent 6,704,925. Claims 12-15, 18 and 19 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Bugnion in view of Yates et al., (Yates), U.S. Patent 5,802,373. Claims 7 and 9 stand objected-to as containing the informalities of words being broken internally by spaces.

The Applicants have not amended objected-to claims 7 and 9 to correct the noted informalities (words broken internally by spaces). The informalities do not appear in the originally filed application document, so it seems likely that these informalities were added during the scanning process at the USPTO. The Applicants therefore respectfully request that the informalities be corrected without requiring an amendment by Applicants.

Support for the amendments to claims 9 and 20 and for new claim 23 is found at least in FIG. 3, page 24, lines 4-16 and page 26, lines 5-17 of Applicants' specification. Claim 18 has been amended to correct a grammatical error. No new matter has been entered.

The Applicants believe that the currently pending claims are not anticipated by or obvious over the cited references and respectfully request reconsideration.

Claim 1

The cited references do not disclose or suggest:

"Apparatus for dynamically transforming and caching at least one computer program, the apparatus comprising:

a. one or more computer readable storage media; and

b. computer executable instructions stored in the one or more computer readable storage media, the computer executable instructions comprising:

i. instructions for dynamically transforming code fragments;

ii. instructions for caching said code fragments;

iii. instructions for causing said code fragments to be executed by at least one computer processor; and

iv. **instructions providing an application programming interface enabling said at least one computer program to activate said instructions for dynamically transforming said code fragments and said instructions for caching said code fragments.**"

(Claim 1, emphasis added)

At least the above highlighted features which differentiate embodiments of the present invention from the cited references are features that are not anticipated by the cited references and would not have been obvious to a person with ordinary skill in the art having the cited references. For example, Bugnion discloses a virtual machine monitor including a virtual operating system, on which one or more applications run and are translated. (Bugnion, col. 6, lines 14-33) Bugnion does not provide an API enabling a computer program being dynamically transformed and cached to "activate said instructions for dynamically transforming said code fragments and said instructions for caching said code fragments". Bugnion discloses that the virtual machine and its virtual operating system receive the instructions and the binary translation subsystem translates the instructions. (Bugnion, col. 6, lines 26-30) For clarification on the API and on the differences between accessing instructions from a program being transformed using an API and a transparent mechanism, see Applicants' specification, page 10, line 32 - page 11, line 5 as follows:

> The API 22 exports functions accessing the caching and linking to the application, enabling explicit control of the core 20 over the execution. The transparent mode layer 24 enables the core 20 to transparently gain control over the execution, that is, the application 12 has no indication that its execution is being controlled by the DELI 10 when it runs through the transparent mode layer 24.

Clearly, Bugnion does not disclose an API as in claim 1 above, but discloses a transparent control only via its virtual operating system.

Applicants believe that claim 1 is allowable over the cited references for at least the reasons set forth above and respectfully request reconsideration.

Claim 2 is believed allowable as depending from an allowable base claim and is further believed allowable in that the cited references do not disclose or suggest:

> "The apparatus of claim 1, **wherein said instructions providing an application programming interface enable said at least one computer program to provide said code fragments for said instructions for dynamically transforming code fragments and for said instructions for caching said code fragments.**"

(Claim 2, emphasis added)

Applicants repeat the arguments set forth above with respect to claim 1. As discussed above, the cited references do not disclose an API enabling a computer program being transformed or cached to provide code fragments. Rather, Bugnion discloses transparently obtaining instructions using a virtual operating system.

Dependent claims 3-8 depend upon independent claim 1 which is allowable over the cited art as discussed above. These dependent claims are likewise in condition for allowance at least because they depend on an allowable independent claim. However, dependent claims 3-8 are independently allowable at least in that they recite particular features which, when combined with the elements of the independent claim, are also not disclosed or suggested in the cited references.

Claim 9 is believed allowable as depending from an allowable base claim and is further believed allowable in that the cited references do not disclose or suggest:

> "The apparatus of claim 1, wherein said at least one computer program comprises **a plurality of emulators being executed simultaneously.**"
> (Claim 9, as amended, emphasis added)

Although Bugnion does disclose incorporating a dynamic binary translator into an emulator (Bugnion, col. 9, lines 46-48), and does disclose executing one or more applications programs on a virtual operating system in a dynamic binary translator, the Applicants respectfully disagree that Bugnion discloses or suggests simultaneously executing a plurality of emulators on an apparatus for dynamically transforming and caching computer programs. One exemplary embodiment of the dynamic execution layer interface disclosed in Applicants' specification provides modularity and code caching functions to multiple different emulators running simultaneously on a processor, simplifying the emulators and easing transitions to different target hardware. (Applicants' specification, page 25, line 27 - page 26, line 7) Thus, software written for a variety of platforms could be executed simultaneously on a single processor through the Applicants' claimed apparatus. Bugnion only discloses including a virtual machine in a single architecture emulator or simulator. Bugnion discloses only a single emulator and therefore cannot provide these benefits.

Claim 10 is believed allowable as depending from an allowable base claim and is further believed allowable in that the cited references do not disclose or suggest:

> "The apparatus of claim 9, **wherein said plurality of emulators comprise emulators for at least two different computer architectures.**"
>
> (Claim 10, emphasis added)

Applicants repeat the arguments set forth above with respect to claim 9. As discussed above, Bugnion does not disclose the simultaneous transformation and execution of a plurality of emulators, particularly emulators for different computer architectures.

Dependent claims 11-17 depend ultimately upon independent claim 1 which is allowable over the cited art as discussed above. These dependent claims are likewise in condition for allowance at least because they depend on an allowable independent claim. However, dependent claims 11-17 are independently allowable at least in that they recite particular features which, when combined with the elements of the independent claim, are also not disclosed or suggested in the cited references.

Claim 18 is believed allowable as depending from an allowable base claim and is further believed allowable in that the cited references do not disclose or suggest:

> "The apparatus of claim 1, said computer executable instructions further comprising **instructions for replacing hardware control code in said code fragments, where said hardware control code is adapted to control** hardware which is not present and **hardware which is not functioning.**"
>
> (Claim 18, as amended, emphasis added)

Applicants concede that conventional translators, likely including those disclosed by Bugnion and Yates, inherently replace hardware control code that is adapted to control hardware which is not present, specifically, the processor for which the code fragments were originally written. Applicants note, however, that the cited references do not disclose or suggest software emulation of unavailable hardware such as peripherals or computer components other than the processor, as described in Applicants' specification at page 6, lines 25-33. Furthermore, the cited references do not disclose or suggest instructions for replacing hardware control code that is adapted to control non-functioning hardware. The Examiner has cited Yates, col. 1, lines 15-18, which states:

> The operating system includes all of the software resources needed by the computer system to interface each of the hardware elements to the computer system...

Applicants respectfully note that this is simply a description of a conventional operating system's function of interfacing computer hardware, not instructions for replacing hardware control code for missing or non-functioning hardware.

As neither cited reference, taken singly or in combination, discloses or suggests instructions for replacing hardware control code that is adapted to control non-functioning hardware, a *prima facie* case of obviousness has not been established. MPEP Section 706.02(j) sets forth the following regarding the establishment of a *prima facie* case:

> To establish a prima facie case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, **the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on applicant's disclosure.**

Applicants believe that currently pending claim 18 is allowable over the cited references for at least the reasons set forth above and respectfully request reconsideration.

<u>Claim 19</u>

The cited references do not disclose or suggest:

"Apparatus for dynamically transforming and caching at least one computer program, the apparatus comprising:

a. one or more computer readable storage media; and

b. computer executable instructions stored in the one or more computer readable storage media, the computer executable instructions comprising:

i. instructions for dynamically transforming code fragments;

ii. instructions for dynamically optimizing said code fragments;

iii. instructions for caching said code fragments;

iv. instructions for changing hardware control code in said code fragments;

v. instructions for transparently obtaining said code fragments from said at least one computer program;

vi. **instructions providing an application programming interface enabling said at least one computer program to activate said instructions for dynamically transforming and caching said code fragments**; and

vii. instructions for causing said code fragments to be executed by at least one computer processor."

(Claim 19, emphasis added)

Applicants repeat the arguments set forth above with respect to claim 1, but specifically directed to the apparatus of claim 19.

<u>Claim 20</u>

The cited references do not disclose or suggest:

"Apparatus for executing a plurality of software applications, the apparatus comprising:

a. one or more computer readable storage media; and

b. computer executable instructions stored in the one or more computer readable storage media, the computer executable instructions comprising:

i. instructions for obtaining portions of computer program code from said plurality of software applications, **said plurality of software applications comprising emulators;**

ii. instructions for dynamically transforming and caching said portions of computer program code to create transformed code fragments; and

iii. instructions for executing said transformed code fragments, **wherein a single instantiation of said computer executable instructions causes said executing of said transformed code fragments from said plurality of software applications simultaneously.**"

(Claim 20, as amended, emphasis added)

Applicants repeat the arguments set forth above with respect to claims 9, but specifically directed to the apparatus of claim 20.

Claim 22 is believed allowable as depending from an allowable base claim and is further believed allowable in that the cited references do not disclose or suggest:

"The apparatus of claim 21, **wherein said plurality of software applications emulate computer systems with at least two different instruction set architectures.**"

(Claim 22, emphasis added)

Applicants repeat the arguments set forth above with respect to claim 9. As discussed above, Bugnion does not disclose the simultaneous transformation and execution of a plurality of emulators, particularly emulators for different computer architectures.

Claim 23 is believed allowable as depending from an allowable base claim and is further believed allowable in that the cited references do not disclose or suggest:

> "The apparatus of claim 9, **wherein said plurality of emulators are executed simultaneously on said at least one computer processor using a single instantiation of said instructions for dynamically transforming code fragments.**"
>
> (Claim 23, emphasis added)

Applicants repeat the arguments set forth above with respect to claim 9. Furthermore, the cited references do not disclose or suggest that the plurality of emulators are executed simultaneously using a single instantiation of instructions for dynamically transforming code fragments.

The Applicants believe that the currently pending claims are allowable over the cited references and respectfully request the timely issuance of a Notice of Allowance.

Dated: 11/4/04

Respectfully submitted,
KLAAS, LAW, O'MEARA & MALKIN, P.C.

By: GK Clinger
Guy K. Clinger, Esq.
Registration No. 42,422
1999 Broadway, Suite 2225
Denver, CO 80202
(303) 298-9888
Fax: (303) 297-2266